MCP SERVER | NO CODE | CLOUD HOSTED

# Grafana MCP for AI Agents

## Manage your observability and monitoring dashboards through natural language.

Grafana MCP lets your AI agent manage your observability stack. You can search for dashboards, inspect specific queries, audit your data sources, and check active alerts without leaving your chat interface. It bridges the gap between your monitoring data and your agent's reasoning.

A+ Quality Score 100/100

dashboards | metrics | logs | traces | alerting | data-visualization

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JULY 2026
vinkius.com/mcp/grafana

SCAN TO VIEW

# The connectivity layer between AI and the world's software.

Vinkius sits between AI and every application. All communication passes through Vinkius Cloud via the Model Context Protocol (MCP) — with governance, observability, and security at every layer.

# Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the connectivity layer where AI connects to the software your business already runs. We handle the hosting, the security, the credentials, the uptime — you get agents that actually do things.

We operate the world's largest managed MCP catalog. Major SaaS platforms, CRMs, databases, and cloud providers — running, monitored, production-ready. This MCP server is hosted and maintained by the Vinkius Cloud for AI Agents.

> *The agent doesn't manage credentials, doesn't manage uptime, doesn't manage security. Vinkius does.*
>
> — Architecture principle

## Four Pillars of the Vinkius Runtime

**01 — Security by design**

Credentials stay encrypted at rest via AES-256. The AI agent never touches raw keys — they're injected into a sandboxed V8 isolate at runtime. Actions are logged, and connections have an emergency kill switch.

**02 — Built on MCP Fusion**

This MCP server was built with **MCP Fusion**, the open-source framework (Apache 2.0) that powers the entire Vinkius catalog. Schema-as-firewall strips undeclared fields, compiled PII redaction runs at zero overhead, and cryptographic lockfiles produce git-diffable audit trails.

**03 — Deterministic observability**

Eight immutable metrics per endpoint: request volume, p95 latency, error rate, active connections, cost attribution. A live payload feed logs every tool call with mutation detection.

**04 — Autonomous operations**

Servers are deployed, monitored, and patched autonomously. New capabilities and security patches ship weekly. Zero-downtime deployments ensure continuous availability across all managed MCP servers.

| AES-256 | Ed25519 | 24h TTL | V8 Isolate |
|---|---|---|---|
| Encryption at rest | PKI vault signatures | Ephemeral session keys | Sandboxed execution |

## One Token. Instant Access.

Every MCP server on Vinkius is accessed through a **Connection Token**. Tokens are generated in the cloud dashboard and produce a unique MCP endpoint URL. Paste this URL into any MCP-compatible client — no SDK required.

A single token can serve **multiple AI clients simultaneously**, or you can issue separate tokens per client for granular access control. Each token tracks its own request count, last activity timestamp, and can be individually enabled or revoked.

MCP ENDPOINT

```
https://edge.vinkius.com/{token}/mcp
```

Claude · Cursor · VS Code · Windsurf · Grok · Gemini

## Security Is the Architecture

Security in Vinkius is not a feature — it's the foundation of the runtime. The gateway enforces multiple independent protection layers between AI agents and third-party APIs.

**01 — Ed25519 PKI Vault**

Every workspace has an Ed25519 Master Key. Session keys are generated ephemerally (24h TTL) and signed by the Master Key. Credentials never leave the vault boundary.

**02 — V8 Isolate Sandboxing**

Tool code runs inside isolated-vm V8 isolates with 64 MB memory caps and per-request timeouts. No filesystem access, no network access except through the SSRF-guarded fetch bridge.

**03 — SSRF Guard**

All outbound HTTP requests are DNS-resolved and validated before execution. Private IP ranges (10.x, 172.16-31.x, 192.168.x, AWS metadata 169.254.x) are blocked at the network layer.

**04 — DLP & PII Redaction**

A ResponseGuard pipeline intercepts every tool response. Configurable redaction patterns strip sensitive fields (emails, SSNs, card numbers) before data reaches the AI agent.

**05 — Cryptographic Audit Trail**

Every request is signed into a SHA-256 hash chain with Ed25519 signatures. Events form a tamper-proof, SIEM-exportable forensic record.

**06 — Honeytoken Trap System**

Phantom credentials are injected into isolated environments. If a honeytoken is used outside Vinkius infrastructure, the server is quarantined instantly.

# Emergency Kill Switch

EU AI Act Art. 14(1) Compliant

The kill switch is an **emergency halt** mechanism — not a simple toggle. When triggered, it executes three actions atomically:

**01 — Server deactivated**

The MCP server is immediately taken offline across the entire cluster.

**02 — All tokens revoked**

Every connection token is invalidated. Total lockout — reconnection blocked until new tokens are issued.

**03 — WebSocket connections killed**

Active connections terminated via Redis pubsub broadcast. Propagates to every runtime node in the cluster.

## Full Visibility. Zero Guesswork.

The Vinkius cloud dashboard includes a full MCP Governance suite — real-time analytics and security controls for production AI operations.

**Control Plane**

KPI dashboard with request volume, latency, success rate, token consumption, and AI-generated operational briefings.

**Agent Activity**

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

**FinOps**

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

**Tool Health**

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

**Firewall & DLP**

PII redaction activity, sensitive data protection counters, and security event timeline.

**Incident Log**

Error trends, failure rates, status-code breakdowns, and forensic audit trail access.

> Get started at **cloud.vinkius.com** — connect your AI agent in under 60 seconds.

# Grafana MCP

4 tools available
Cloud-hosted on Vinkius

When an incident hits at 3 a.m., the last thing you want to do is hunt through a messy list of dashboards to find the right set of metrics. This MCP changes that by giving your AI agent direct eyes on your Grafana instance. Instead of clicking through menus to find a specific PromQL query or checking if a Loki log is actually firing, you just ask your agent to find it for you. It pulls the actual configuration of your panels so you can see exactly what's being measured. You can also use it to audit your data sources to make sure your Prometheus or SQL connections are actually healthy. It's a way to turn your observability stack into a searchable knowledge base. By using this on the Vinkius marketplace, you get a reliable way to connect your monitoring data to your daily workflow. You'll stop wasting time on manual navigation and start getting answers about your system health in seconds. You can quickly pull out PromQL strings to use in your own scripts or check the status of every active alert rule without refreshing your browser. It makes the entire observability stack feel like a conversation rather than a series of tabs.

## Core Capabilities

**01 — Find dashboards by title or tag**

Search your Grafana instance to find the exact dashboard you need without manual browsing.

**02 — Pull full panel configurations**

Retrieve the complete layout and query details for any dashboard using its unique ID.

**03 — List all active data sources**

Get a complete list of all connected data sources like Prometheus, Loki, and SQL.

**04 — Check currently firing alerts**

See a real-time list of alerts that are currently in a firing state with all relevant labels.

**05 — Extract PromQL and SQL queries**

Get the exact query strings from your panels to use in your own debugging or scripts.

**06 — Audit monitoring variables**

Analyze localized variables and structural constraints across your monitoring environment.

## One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/grafana** — connect your AI agent in three steps.

01 Subscribe to the MCP and grab your Service Account Token from your Grafana settings.

02 Provide your Grafana URL and token to your AI client.

03 Ask your agent to find dashboards, pull queries, or check alert statuses.

The bottom line is you get instant access to your monitoring data through natural language.

## Built For

This is for the SREs and DevOps engineers who are tired of clicking through dozens of tabs during a high-pressure incident. It's for anyone who needs to move from manual dashboard navigation to instant, query-based answers.

**SRE Engineer**

Checks firing alerts and pulls PromQL queries during active incidents to identify root causes faster.

**DevOps Engineer**

Audits data source connections and searches for specific dashboards to verify deployment health.

**Cloud Architect**

Verifies data source configurations and dashboard organization across different environments.

**Software Developer**

Extracts specific queries from production panels to verify metrics during the development cycle.

## What Changes When You Connect

01 Stop manual dashboard hunting by using search_dashboards to find the exact metrics you need in seconds.

02 Get precise PromQL and SQL queries instantly with get_dashboard so you don't have to dig through the UI.

03 Verify your entire stack's health by using list_datasources to audit Prometheus, Loki, and SQL connections.

04 Reduce incident response time by using firing_alerts to see exactly what's broken without refreshing the browser.

05 Simplify your development workflow by letting your agent extract panel layouts and variables directly from your environment.

## Real-World Applications

**Rapid Incident Response**

An alert goes off and you ask your agent to find the 'Production API' dashboard and check if any other alerts are firing to see the full scope of the issue.

**Query Extraction for Devs**

A developer needs a specific PromQL query from a dashboard. They ask the agent to pull the dashboard config and copy the query for use in a script.

**Infrastructure Audit**

A cloud architect needs to know every data source in the Grafana instance. They ask the agent to list them all to verify security boundaries.

**Dashboard Discovery**

A new team member needs to find all dashboards tagged with 'billing'. They ask the agent to list them by tag to get oriented quickly.

## Patterns to Avoid

### Searching for everything at once

✗ AVOID

Asking the agent to show every single dashboard and query in the system in one go.

✓ INSTEAD

Use search_dashboards first to find the specific dashboard you need, then use get_dashboard to pull the details for that specific UID.

### Manual Query Copying

✗ AVOID

Navigating to a dashboard and manually highlighting a PromQL query to copy it.

✓ INSTEAD

Use get_dashboard to have the agent pull the exact query string for you automatically.

### Ignoring Alert Context

✗ AVOID

Looking at a firing alert but not knowing what other rules are affected.

✓ INSTEAD

Use firing_alerts to get a comprehensive list of all active issues in one command.

## The Right Fit

Use this if you need your AI agent to interact with your Grafana instance to pull metrics, check alert statuses, or audit dashboard configurations. It's built for people who want to move from clicking to asking. It is ideal for SREs who need to know what's wrong during an outage or for developers who need to quickly grab a query from a production panel. Don't use this if you just want to visualize data in a browser or if you need to build new dashboards. This MCP is for reading and inspecting what already exists in your environment. If you need to create new alerts or build new panels, you still need to use the Grafana UI. This tool is your shortcut to the data, not a replacement for the dashboard builder.

# Grafana MCP for Observability and Incident Response

When an incident hits at 3 a.m., the last thing you want to do is hunt through a messy list of dashboards to find the right set of metrics. You're clicking through tabs, searching for specific UIDs, and copy-pasting PromQL queries from the browser while the clock is ticking.

With this MCP, your agent does the digging for you. You ask a question, and it pulls the dashboard config or alert status directly into your chat. You get immediate answers instead of manual navigation through multiple tabs.

# Grafana MCP for Infrastructure Auditing and Query Extraction

Manually auditing every data source connection, verifying permissions, and listing out every active alert in a spreadsheet is a tedious way to manage a stack. It's slow and prone to human error.

This MCP automates the audit. Your agent can list all data sources or fetch firing alerts in one go. You get a clear picture of your system health without the tedious manual work.

# Grafana MCP 4 Tools for Observability

Use these tools to search dashboards, pull queries, list data sources, and check active alerts through your AI agent.

| # | TOOL | DESCRIPTION |
|---|---|---|
| 01 | `search_dashboards` | Find specific dashboards using titles or tags. It returns the unique IDs you need for deeper inspection. |
| 02 | `get_dashboard` | Pull the full configuration of a dashboard using its UID. This lets you see every panel and query in the set. |
| 03 | `list_datasources` | See every data source connected to your Grafana instance. It's the fastest way to verify your connectivity boundaries. |
| 04 | `firing_alerts` | Get a list of alerts that are currently in a firing state. You can see labels and annotations for active incidents. |

# See It in Action

Real prompts you can use once this MCP is connected to your AI agent through Vinkius Cloud.

**U** Are there any active alerts right now?

Checking your alert rules... I found 1 firing alert: **High Error Rate** on service **API-v1**. It has been firing for 15 minutes. All other 12 alert rules are currently in 'Normal' state.

**U** Find the production billing dashboard and show me the queries.

I found the **Production Billing** dashboard (UID: `bill-99`). It contains 4 panels. The **Success Rate** panel uses this PromQL:

```
sum(rate(http_requests_total{status=~"2..3"}[5m])) / sum(rate(http_requests_total[5m]))
```

**U** List all our data sources.

Your Grafana instance is connected to the following 5 data sources:

- **Prometheus** (Primary Metrics)
- **Loki** (Logs)
- **CloudWatch** (AWS Metrics)
- **PostgreSQL** (User Data)
- **InfluxDB** (IoT Sensors)

# Frequently Asked Questions

01 **Can the Grafana MCP help me during a production outage?**

Yes, it lets your agent check firing alerts instantly so you can see what's broken without hunting through the UI.

02 **Can I use the Grafana MCP to see my PromQL queries?**

Yes, it can pull the full configuration of any dashboard, including all the underlying queries for every panel.

03 **Does the Grafana MCP support Loki or SQL databases?**

Yes, it can list all configured data sources, which includes Loki, SQL, and Prometheus.

04 **How does the Grafana MCP find specific dashboards?**

You can ask your agent to search by title or tag, and it will find the right one for you automatically.

05 **Can I use this to audit my monitoring setup?**

Yes, it's great for listing all data sources and checking alert rules to ensure everything is configured correctly.

06 **Is the Grafana MCP good for developers?**

It's excellent for devs who need to quickly grab metrics or log queries from existing production dashboards.

07 **Can my agent search for specific dashboards in my Grafana instance?**

Yes. Use the 'search_dashboards' tool. You can provide an optional query string to match titles or tags. The agent will return basic info including the unique UID required for deeper inspection.

08 **How do I extract the PromQL or SQL queries from a dashboard panel via chat?**

Use the 'get_dashboard' tool with the dashboard UID. Your agent will retrieve the full JSON configuration, including all panels and their underlying data queries, enabling you to review the exact metrics logic natively.

09 **Can I see firing alerts through the agent?**

Absolutely. Use the 'list_alerts' tool. The agent retrieves all configured alert rules and their current statuses, allowing you to identify which monitors are currently in a firing state synchronousy.

# Go Live in 60 Seconds

Get your connection token from **cloud.vinkius.com**, then paste the endpoint URL into any MCP-compatible client.

YOUR MCP ENDPOINT

```
https://edge.vinkius.com/[TOKEN]/mcp
```

| CLIENT | WHERE TO CONFIGURE |
|---|---|
| Claude AI | Profile → Customize → Connectors → "+" → Add custom connector → Paste endpoint |
| Cursor | Settings → Features → MCP Servers → "+ Add New MCP Server" → Type: SSE → Paste endpoint |
| VS Code | Ctrl/Cmd+Shift+P → "MCP: Add Server" → add `"grafana": { "url": "…" }` |
| Windsurf | MCP Settings → `mcp_settings.json` → Add endpoint URL |
| ChatGPT | Settings → Tools & plugins → Add MCP server → Paste endpoint |
| Gemini | Extensions → Add MCP Server → Paste endpoint URL |

ASK AN AI ABOUT THIS

Let your preferred AI explain this MCP server

- Ask ChatGPT
- Ask Claude
- Ask Perplexity
- Ask Gemini
- Ask Grok

READY TO CONNECT

# Grafana is live on Vinkius Cloud.

Get your connection token, paste it into your AI agent, and start building. No SDK. No deployment. Just results.

**Start at cloud.vinkius.com →**

vinkius.com · support@vinkius.com

## INDEPENDENT PLATFORM DISCLAIMER

Vinkius is an independent platform and is not affiliated with, endorsed by, sponsored by, verified by, or otherwise authorized by Grafana. All third-party trademarks, logos, and brand names are the property of their respective owners. Their use in this document is strictly for informational purposes to identify service compatibility and interoperability.

## DOCUMENT INFORMATION

| | |
|---|---|
| Generated | July 2026 |
| MCP Server | Grafana MCP |
| Server ID | `019d75aa-7428-7281-bf68-5d4097fc9cae` |
| Platform | Vinkius Cloud for AI Agents |
| Endpoint | `https://edge.vinkius.com/{token}/mcp` |

## LICENSE & USAGE

This document is generated automatically by the Vinkius PDF Engine. Content reflects the MCP server configuration at the time of generation and may change as updates are deployed. For the most current information, visit vinkius.com/mcp/grafana.